

Robotic Exploration: Place Recognition as a Typicality Problem

E. Jauregi¹, I. Irigoien¹, E. Lazkano¹, B. Sierra¹ and C. Arenas²

¹*Department of Computer Sciences and Artificial Intelligence,
University of Basque Country*

²*Department of Statistics, University of Barcelona
Spain*

1. Introduction

Autonomous exploration is one of the main challenges of robotic researchers. Exploration requires navigation capabilities in unknown environments and hence, the robots should be endowed not only with safe moving algorithms but also with the ability to recognise visited places. Frequently, the aim of indoor exploration is to obtain the map of the robot's environment, i.e. the *mapping* process. Not having an automatic mapping mechanism represents a big burden for the designer of the map because the perception of robots and humans differs significantly from each other. In addition, the *loop-closing* problem must be addressed, i.e. correspondences among already visited places must be identified during the mapping process.

In this chapter, a recent method for topological map acquisition is presented. The nodes within the obtained topological map do not represent single locations but contain information about areas of the environment. Each time sensor measurements identify a set of landmarks that characterise a location, the method must decide whether or not it is the first time the robot visits that location. From a statistical point of view, the problem we are concerned with is the typicality problem, i.e. the identification of new classes in a general classification context. We addressed the problem using the so-called INCA statistic which allows one to perform a typicality test (Irigoien & Arenas, 2008). In this approach, the analysis is based on the distances between each pair of units. This approach can be complementary to the more traditional approach units \times measurements – or features – and offers some advantages over it. For instance, an important advantage is that once an appropriate distance metric between units is defined, the distance-based method can be applied regardless of the type of data or the underlying probability distribution.

We describe the theoretical basis of the proposed approach and present extensive experimental results performed in both a simulated and a real robot-environment system. Behaviour Based philosophy is used to construct the whole control architecture. The developed system allows the robot not only to construct the map but also comes in useful for localisation purposes.

2. Literature review

Loop-closing has long been identified as a critical issue when building maps from local observations. Topological mapping methods isolate the problem of how loops are closed from the problem of how to determine the metrical layout of places in the map and how to deal with noisy sensors.

The loop-closing problem cannot be solved neither relying only on exteroceptive information (due to sensor aliasing) nor on proprioceptive information (cumulative error). Both environmental properties and odometric information must be used to disambiguate locations and to correct robot position. Fraundorfer et al. (2007) present a highly scalable vision based localisation and mapping method that uses image collections, whereas Se et al. (2005) use vision mainly to detect the so called *loop-closing* –the place has already been visited by the robot– in robot localisation; Tardós et al. (2002) introduce a perceptual grouping process that permits the robust identification and localisation of environmental features from the sparse and noisy sonar data. On the other hand, the probabilistic Bayesian inference, along with a symbolic topological map is used by Chen & Wang (2006) to relocalise a mobile robot. More recently, Olson (2009) presents a new loop-closing approach based on data association, where places are recognised by performing a number of pose-to-pose matchings; a review of loop-closing approaches related to MONOSLAM can be found in (Williams et al., 2009). Within the field of probabilistic robotics (Thrun et al., 2005), Kalman filters, Bayesian Networks and particle filters are used to maintain probability distributions over the state space while solving mapping, localisation and planning.

But the mapping problem can also be stated from a classification perspective. In most classification problems, there is a training data available for all classes of instances that can occur at prediction time. In this case, the learning algorithm can use the training data to determine decision boundaries that discriminate among the classes. However, there are some problems that exhibit only a single class of instances at training time but are amenable to machine learning. At prediction time, new instances with unknown class labels can either belong to the target class or to a new class that was not available during training. In this scenario, two different predictions are possible: *target*, an instance that belongs to the class learnt during training, and *unknown*, where the instance does not seem to belong to the previously learnt class. Within the machine learning community, this kind of problems are known as *one-class* problems and as *typicality* problems within the statistics research.

To give some examples, in (Hempstalk et al., 2008) the probability distributions of the class variable known values are used to determine if a new case belongs to the known class values or if it should be considered as a different class member. One-class classification categorizers have a wide range of applications; in (Manevitz & Yousef, 2007) one-class classification is used to document categorisation in order to decide whether a reference is relevant in a database searching query. The same authors combine this approach with the Support Vector Machine (SVM) paradigm for document classification purposes (Manevitz & Yousef, 2002); and in (Sánchez-Yáñez et al., 2003) the same idea is applied to texture recognition in images. A thorough review of one-class classification can be found in (Tax, 2001).

Regarding the mobile robotics area, one-class classification approaches can be applied to robot mapping, i.e. to learn the structure of its environment in an automatic manner. In this way, Brooks & K. Iagnemma (2009) present a use of this approach to deal with terrain recognition, and Wang & Lopes (2005) use it to identify user actions in human-robot-interaction. However,

direct uses of this approach, with this particular name, have not been found in the robotics literature.

There are different approaches found in the literature to deal with the typicality problem (Bar-Hen, 2001; Cuadras & Fortiana, 2000; Irigoien & Arenas, 2008; McDonald et al., 1976; Rao, 1962). Some of them are only suitable for normal multivariate data, others are appropriate for any kind of data but are limited to $k = 2$, being k the number of classes. The latter case offers the most general framework to be applied. However, and in spite of the high diversity of the used methods, to the best of the author's knowledge, neither typicality nor one-class approaches appear in the mapping literature.

The approach proposed in this chapter combines the INCA statistic (Irigoien & Arenas, 2008) with the topological properties of the environmental locations considered and thus represents a new approach to tackling the robot mapping problem as a typicality case.

3. Typicality test by means of the INCA statistic

In this section the INCA statistic is introduced and the INCA test is proposed as a solution to the typicality problem.

3.1 Preliminaries

The data we consider are random vectors and we assume that distinct classes exist. Let C_1, C_2, \dots, C_k be k classes represented as k independent S -valued random vectors $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_k$, with probability density functions f_1, f_2, \dots, f_k with respect to a suitable common measure λ . Let $\delta(\mathbf{y}, \mathbf{y}')$ be a distance (Gower, 1985) function on S . We say that δ is a Euclidean distance function if the metric space (S, δ) can be embedded in a Euclidean space, $\Psi : S \rightarrow \mathbb{R}^p$, such that:

$$\delta^2(\mathbf{y}, \mathbf{y}') = \|\Psi(\mathbf{y}) - \Psi(\mathbf{y}')\|^2, \quad (1)$$

and we may understand $E(\Psi(\mathbf{Y}_i))$ as the δ -mean of $\mathbf{Y}_i, i = 1, \dots, k$.

In this general framework the following concepts are considered. The geometric variability of $C_i, i = 1, \dots, k$ with respect to δ is defined (Cuadras & Fortiana, 1995) as

$$V_\delta(C_i) = \frac{1}{2} \int_{S \times S} \delta^2(\mathbf{y}_{i1}, \mathbf{y}_{i2}) f(\mathbf{y}_{i1}) f(\mathbf{y}_{i2}) \lambda(d\mathbf{y}_{i1}) \lambda(d\mathbf{y}_{i2}).$$

This quantity is a variant of Rao's diversity coefficient (Rao, 1982). When δ is the Euclidean distance and $\Sigma_i = \text{COV}(\mathbf{Y}_i)$, then $V_\delta(C_i) = \text{tr}(\Sigma_i)$. For other dissimilarities $V_\delta(C_i)$ is a general measure of dispersion of \mathbf{Y}_i . In the context of discriminant analysis (Cuadras et al., 1997) the squared distance between C_i and C_j is defined by

$$\Delta^2(C_i, C_j) = \int_{S \times S} \delta^2(\mathbf{y}_i, \mathbf{y}_j) f(\mathbf{y}_i) g(\mathbf{y}_j) \lambda(d\mathbf{y}_i) \lambda(d\mathbf{y}_j) - V_\delta(C_i) - V_\delta(C_j) \quad (2)$$

This quantity is the Jensen difference (Rao, 1982) between the distributions of C_i and C_j . If the metric space (S, δ) can be embedded (see (1)) in a Euclidean space \mathbb{R}^p and if $E(\|\Psi(\mathbf{Y}_i)\|)$ and $E(\|\Psi(\mathbf{Y}_i)\|^2)$ are finite, then $V_\delta(C_i) = E(\|\Psi(\mathbf{Y}_i)\|^2) - \|E(\Psi(\mathbf{Y}_i))\|^2, i = 1, \dots, k$, and $\Delta^2(C_i, C_j) = \|E(\Psi(\mathbf{Y}_i)) - E(\Psi(\mathbf{Y}_j))\|^2$. If there is only one element $C_i = \{\mathbf{y}_0\}$, (3) gives the proximity function of \mathbf{y}_0 to C_j ,

$$\phi^2(\mathbf{y}_0, \mathbf{Y}_j) = \int_S \delta^2(\mathbf{y}_0, \mathbf{y}_j) f(\mathbf{y}_j) \lambda(d\mathbf{y}_j) - V_\delta(C_j). \quad (3)$$

In applied problems the distance function is typically a datum, but the probability distribution for each population is unknown. Natural estimators given samples $\mathbf{y}_1^1, \dots, \mathbf{y}_{n_1}^1, \dots, \mathbf{y}_1^k, \dots, \mathbf{y}_{n_k}^k$ of sizes n_1, \dots, n_k coming from C_1, \dots, C_k are the following:

- The *geometric variability* of C_j ,

$$\hat{V}_\delta(C_j) = \frac{1}{2n_j^2} \sum_{l,m} \delta^2(\mathbf{y}_l^j, \mathbf{y}_m^j).$$

- The *proximity function* of a new object \mathbf{y}_0 to C_j ,

$$\hat{\phi}^2(\mathbf{y}_0, C_j) = \hat{\phi}_j^2(\mathbf{y}_0) = \frac{1}{n_j} \sum_l \delta^2(\mathbf{y}_0, \mathbf{y}_l^j) - \hat{V}_\delta(C_j).$$

- The *squared distance* between C_i and C_j ,

$$\hat{\Delta}^2(C_i, C_j) = \hat{\Delta}_{ij}^2 = \frac{1}{n_i n_j} \sum_{l,m} \delta^2(\mathbf{y}_l^i, \mathbf{y}_m^j) - \hat{V}_\delta(C_i) - \hat{V}_\delta(C_j). \quad (4)$$

See (Arenas & Cuadras, 2002) and references therein for a review of these concepts, their application, different properties and proofs.

3.2 INCA statistic

Consider that n units are simply divided into k classes C_1, \dots, C_k , of sizes n_1, \dots, n_k . Consider a fixed unit \mathbf{y}_0 , which may be an element of a C_j , $j = 1, \dots, k$ or may belong to an unknown class, i.e. it may be an atypical unit. Consider a new class with δ -mean the linear combination $\sum_{i=1}^k \alpha_i E(\Psi(\mathbf{Y}_i))$, where \mathbf{Y}_i is the random vector representing the class C_i , $i = 1, \dots, k$. The INCA statistic is defined as follows:

$$W(\mathbf{y}_0) = \min_{\alpha_i} \{L(\mathbf{y}_0)\}, \quad \sum_{i=1}^k \alpha_i = 1, \quad (5)$$

$$L(\mathbf{y}_0) = \sum_{i=1}^k \alpha_i \phi_i^2(\mathbf{y}_0) - \sum_{1 \leq i < j \leq k} \alpha_i \alpha_j \Delta_{ij}^2.$$

$\phi_i^2(\mathbf{y}_0)$ is the proximity function of \mathbf{y}_0 to C_i and Δ_{ij}^2 is the squared distance between C_i and C_j . The INCA statistic $W(\mathbf{y}_0) = \min_{\alpha_i} L(\mathbf{y}_0)$ trades off between minimising the weighted sum of proximities of \mathbf{y}_0 to classes (which takes into consideration the within-group variability) and maximising the weighted sum of the squared distances between classes (between-groups variability) - a common behaviour of a classing criterion. The values of $\alpha' = (\alpha_1, \dots, \alpha_{k-1})$ together with $\alpha_k = 1 - \sum_{i=1}^{k-1} \alpha_i$, verifying (5) are $\mathbf{ff}' = M^{-1}N$, where M is the $(k-1) \times (k-1)$ matrix

$$M = \left(\Delta_{ik}^2 + \Delta_{jk}^2 - \Delta_{ij}^2 \right)_{i,j=1,\dots,k-1}$$

and N is the $(k-1) \times 1$ vector

$$N = \left(\Delta_{ik}^2 + \phi_k^2(\mathbf{y}_0) - \phi_i^2(\mathbf{y}_0) \right)_{i=1,\dots,k-1}.$$

The statistic $W(\mathbf{y}_0)$ has a very nice geometric interpretation. It can be interpreted (see Figure 1) as the (squared) orthogonal distance or height h of \mathbf{y}_0 on the hyperplane generated by the δ -mean of C_i ($i = 1, \dots, k$), denoted in Figure 1 by \mathbf{a}_i , $i = 1, \dots, k$. Then, points which lie significantly far from this hyperplane are held to be outliers. This intuitive idea is used to detect outliers among existing classes.

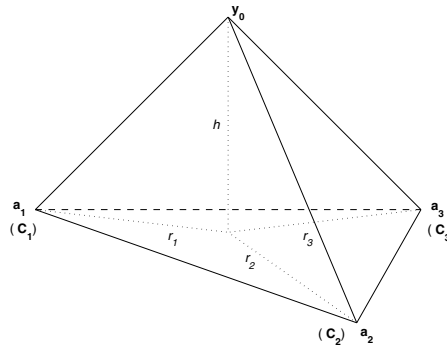


Fig. 1. For $k = 3$, new observation $\{\mathbf{y}_0\}$, centres of classes $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ and (squared) projection \mathbf{r}_i of the edges $\{\mathbf{y}_0, \mathbf{a}_i\}$ on the plane $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$. The (squared) height h is $W(\mathbf{y}_0)$

Suppose now that the data are classified in k classes. Let \mathbf{y}_0 be a new observation and consider the test to decide whether \mathbf{y}_0 belongs to one of the fixed classes C_j , $j = 1, \dots, k$ or, on the contrary, it is an outlier or an atypical observation which belongs to a different and unknown class. Consider the INCA test,

H_0 : \mathbf{y}_0 comes from the class with

$$\delta\text{-mean} \sum_i \alpha_i E(\Psi(\mathbf{Y}_i)), \quad \sum_{i=1}^k \alpha_i = 1, \quad i = 1, \dots, k,$$

H_1 : \mathbf{y}_0 comes from another unknown class,

and compute statistic (5). If $W(\mathbf{y}_0)$ is significant it means that \mathbf{y}_0 comes from a different and unknown class. Otherwise we allocate \mathbf{y}_0 to C_i using the rule:

$$\text{Allocate } \mathbf{y}_0 \text{ to } C_i \text{ if } U_i(\mathbf{y}_0) = \min_{j=1, \dots, k} \{U_j(\mathbf{y}_0)\}, \quad (6)$$

where $U_j(\mathbf{y}_0) = \phi_j^2(\mathbf{y}_0) - W(\mathbf{y}_0)$, $j = 1, \dots, k$.

It can be observed (Irigoin & Arenas, 2008) that $U_j(\mathbf{y}_0)$ represents the (squared) projection of $\{\mathbf{y}_0, E(\Psi(\mathbf{Y}_i))\}$ on the hyper plane $\{E(\Psi(\mathbf{Y}_1)), \dots, E(\Psi(\mathbf{Y}_k))\}$. See Figure 1, where for simplicity the (squared) projection $U_j(\mathbf{y}_0)$ is denoted by r_j , $j = 1, \dots, k$. Hence, criterion 6 follows the next geometric and intuitive allocation rule: Allocate \mathbf{y}_0 to C_i if the projection $U_i(\mathbf{y}_0)$ is the smallest.

We obtained sampling distributions of $W(\mathbf{y}_0)$ and $U_j(\mathbf{y}_0)$ ($j = 1, \dots, k$) by re-sampling methods, in particular drawing bootstrap samples as follows. Draw N units \mathbf{y} with replacement from the union of C_1, \dots, C_k and calculate the corresponding $W(\mathbf{y})$ and

$U_j(\mathbf{y})$ ($j = 1, \dots, k$) values. As usual, this process is repeated $10P$ times with $P \geq 1$ selected by the user. In this way, the bootstrap distributions under H_0 are obtained.

4. Behavior-Based navigation

Behavior-Based (BB) systems appeared in 1986, when R.A. Brooks proposed a bottom-up approach for robot control that imposed a new outlook for developing intelligent embodied agents capable of navigating in real environments performing complex tasks. He introduced the Subsumption Architecture (Brooks, 1986; Brooks & Connell, 1986) and developed multiple robotic creatures capable of showing different behaviours not seen before in real robots (Brooks, 1989; Connell, 1990; Matarić, 1990). Behavior-based systems are originally inspired on biological systems. Even the most simple animals show navigation capabilities with high degree of performance. For those systems, navigation consist of determining and maintaining a trajectory to the goal (Mallot & Franz, 2000). The main question to be answered for navigation is not *Where am I?* but *How do I reach the goal?* and the answer does not always require knowing the initial position. Therefore, the main abilities the agent needs in order to navigate are to move around and to identify goals.

The behavior-based approach to robot navigation relies on the idea that the control problem is better assessed by bottom-up design and incremental addition of light-weight processes, called behaviors, where each one is responsible for reading its own inputs and sensors, and deciding the adequate motor actions. There is no centralized world model and data from multiple sensors do not need to be merged to match the current system state in the stored model. The motor responses of the several behavioural modules must be somehow coordinated in order to obtain valid intelligent behavior. *Way-finding* methods rely on *local navigation strategies*. How these local strategies are coordinated is a matter of study known as *motor fusion* in BB robotics, opposed to the well known *data fusion* process needed to model data information. The aim is to match subsets of available data with motor decisions; outputs of all the active decisions somehow merge to obtain the final actions. In this case there is no semantic interpretation of the data but behavior emergence.

5. Topological places

Generally speaking, there are two typical strategies for deriving topological maps: one is to learn the topological map directly; the other is to first learn a geometric map, then to derive a topological model from it through some process of analysis (Thrun, 1999; Thrun & Bücken, 1996a,b).

As mentioned before, BB systems advocates for a functional bottom-up decomposition of the control problem in independent processes called behaviours. From this point of view, the topological “map” should be composed of tightly coupled behaviours, specific to the meaningful locations.

A topological map is formally defined as a set of nodes where each node consists of:

1. A set of inputs (from landmark identification subsystems) and outputs. These outputs should serve to reduce the distance between the current state and the goal.
2. A signature that identifies the node: *signature_i*. Each location has a signature that reflects the state of a set of specific landmarks and that is used by the robot for localisation purposes.

3. A function α_i to be executed when the node i is active and that will output the action to be performed at the node specific current state. The behaviour of the robot as well as the associated function of the nodes can be different depending on the location.
4. The location identifier that contains initial and final position of the node:

$$(x_{i0}, y_{i0}), (x_{if}, y_{if})$$

The overall “map” is then composed of sets of behaviours, each launched on a different thread. The environment is only partially unknown to the robot since it is provided with behaviour modules to properly identify certain features such as corridors, crossings or junctions and halls, each of them identifiable using distance sensors like a laser scanner. Each landmark identifier outputs a confidence level (cl) as a measure of the confidence of the identification process. These values are filtered through node signatures, giving at each time step the node activation level according to the sensor readings.

- Corridors: the robot is considered to be in a corridor if the place is between 1.6 and 2.4 m wide. To that aim, left and right side shortest readings are summed and stored in a FIFO buffer. The mean of the buffer is used in a Gaussian function that gives the confidence level of being in a corridor.
- Halls: as opposed to corridors, halls are wide areas. Therefore, the confidence level of being in a hall is defined as 1 minus the probability of being in a corridor.
- Crossings or junctions: these locations are areas where two or more alternative ways are possible. It is mandatory for the robot to identify junctions in order to choose the right way when looking for goals. Depending on the destination, the robot must select one way or another. Crossing areas usually come at the end of a corridor or hall and lead to a new area. Hence, left and right minimum distances are looked for and these minimum values are used as reference for searching continuous interval of readings that exceed the minimum values. The orientations of the possible alternative ways at the junctions are registered according to the robot heading provided by the compass sensor and the indexes of the laser scan that define the different intervals, the orientation of the possible alternative ways at the junctions are registered.

The goal of the mapping process is to fill in the nodes with the information that they must contain. More precisely, the contents of the signature and the location identifier. For this aim, during the learning process and depending on the state of the landmark identification subsystems, i.e. the confidence level of the corridor/hall/junction (cl_{corr} , cl_{hall} and cl_{cross}), the following information is given to the INCA test:

- Initial and mean heading values: θ_0, θ_{mean} .
- Initial and final pose obtained by the odometric subsystem. These poses correspond to the position values of the robot when the node signature activates/deactivates: $(x_0, y_0), (x_f, y_f)$.
- Length (previously named as *duration*) of the area calculated using the initial and final pose information: d .
- Number of alternative ways and their associated orientation: num_ways and $\theta_{w_1}, \dots, \theta_{w_{num_ways}}$.

These measurements will constitute the observations of the random vectors \mathbf{Y} considered in the INCA statistic, as represented in Equation 7.

Corridors, Halls:

$$\mathbf{Y} = (\sin(\theta_0), \cos(\theta_0), \sin(\theta_{mean}), \cos(\theta_{mean}), (x_0, y_0), (x_f, y_f), d) \quad (7)$$

Junctions:

$$\mathbf{Y} = (\sin(\theta_0), \cos(\theta_0), \sin(\theta_{mean}), \cos(\theta_{mean}), \theta_{w_1}, \dots, \theta_{w_{num_ways}}, (x_0, y_0))$$

Note that there are two types of measurements: variables type coordinates in meters and variables type orientation in degrees.

The corridors/halls/crosses can differ in their orientation (mean compass value that the robot maintains when going through them in its canonical path). This is why each physical place will correspond to two or more different nodes in the topological map.

6. Proposed approach

The locations the robot must identify are not only single points but areas surrounding these points. Therefore, we propose firstly, a data generation approach to characterise the areas; and secondly, the application of the INCA test.

Let us assume that the robot has recorded the geometric information (see section 5) of k different places C_1, \dots, C_k , all of them of the same type. There is only one \mathbf{y}_i measurement for each place C_i ($i = 1, \dots, k$). However, the place we want to identify topologically is not just a spot but an area or neighbourhood of the recorded measurement \mathbf{y}_i . In order to do so we generate $n_i - 1$ new observations for each place i which will make up the observations corresponding to the place C_i . These new observations are generated as $\mathbf{y}_i^l = \mathbf{y}_i + U(-u, u)$, $l = 2, \dots, n_i$, where $U(-u, u)$ stands for the uniform distribution with parameters $-u$ and u ($u > 0$). Taking into account that the robot records two kinds of variables, metres and degrees, we consider two kinds of values for the parameter of the uniform distribution, let us call them, u_M and u_{DEG} , respectively.

Once the data corresponding to the k classes –places– are generated, and given \mathbf{y}_0 , the information the robot has recorded when he arrives at a new place, the INCA test can be applied and consequently it is possible to decide whether or not \mathbf{y}_0 corresponds to a new place. In case it is decided \mathbf{y}_0 is not a new place, the conclusion is that \mathbf{y}_0 is one of the places C_1, \dots, C_k according to rule (6).

Pearson distance has been used for the calculus of the interdistances $\delta(\mathbf{y}, \mathbf{y}')$ between \mathbf{y} and \mathbf{y}' . The parameter values used during the experimental phase were $n_i = 10$, $u_M = 2$ and $u_{DEG} = 30$. These values were chosen experimentally as explained in (Jauregi et al., 2011).

7. Exploration behaviour

As stated earlier, the mapping process requires an exploration strategy to guide the robot for the terrain inspection. The strategy used in this proposal, the exploration behaviour is a coordination of the local navigation strategies and landmark identification subsystems the robot is endowed with. The proper combination of these behaviours, allow the safe exploration of the environment.

- Two local navigation strategies that are combined in a cooperative manner (weighted sum): balance the free space at both sides of the robot and follow a desired compass orientation (θ_d).
- Landmark identification subsystems that allow the robot to recognise corridors, left/right walls, halls, junctions and dead-ends. These landmarks are used to change robot's desired orientation. To show an example, Figure 2 shows the coordination of the modules for the case where a dead-end is recognised.

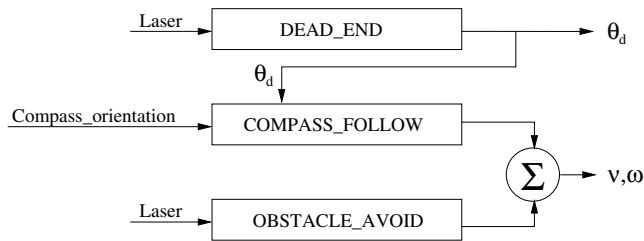


Fig. 2. Diagram of behaviours modules (v : translational velocity, w : angular velocity)

Although the robot can be positioned at any starting location, initially and until the robot reaches a dead-end the map remains empty. Hence, the map construction starts after a dead-end has been identified. This gives the correct measurement of the length of the locations (nodes). Afterwards, the first corridor, the first crossing and the first hall are always identified as new nodes since there is not any instance of the same type already stored in the map.

Once the map building process starts, each time the robot identifies a location – a corridor, a hall or a crossing – the geometric information of the identified location is recorded (the Y vector, Equation 7), and then the INCA test is applied to evaluate if they are locations already visited or new ones. When the location corresponds to a crossing, i.e. a junction, the orientations of the alternative ways the robot can choose are recorded. If the location has been visited before, one of the non-explored paths is randomly selected. In this way, the robot has the chance to cover all the environment. The robot finishes the exploration process when all the alternatives of the crossing nodes have been tried.

8. Simulated experiments

Experiments were carried out in the third floor of the Faculty of Computer Science. This environment is a semi-structured office-like common environment, with regular geometry as can be seen in Figure 3.

The parameter selection obtained in the previous experimental phase was applied to the more general problem of identifying the whole set of environmental locations during an exploration phase performed in simulation. To this purpose the *Stage* simulation tool was used together with the *Player* robot server.

In order to have a wider view of the mapping process, we let the robot move in the environment for a long time (more than 6500 seconds). On the left of the Figure 4 shows the robot's path starting from the dead-end at the bottom left corner and on the right the complete path followed during the exploration of the environment.

		Corr	Hall	Cross
New	Expected	17	8	13
	Found	100%	100%	100%
Known	Traversed	47	23	38
	Classified	100%	100%	100%

Table 1. Experimental results

- Each of the 17 existing corridors were properly labelled as new places the first time the robot went along them; the same happened with the new traversed halls and crossing nodes.
- The nodes visited more than once by the robot in this long journey were also properly classified with their corresponding label; a total number of 47 corridors, 23 halls and 38 crossing nodes were visited in the robot path.

Figure 6 shows the distribution of the locations (plotted according to their central poses) and the evolution of the number of nodes over time.

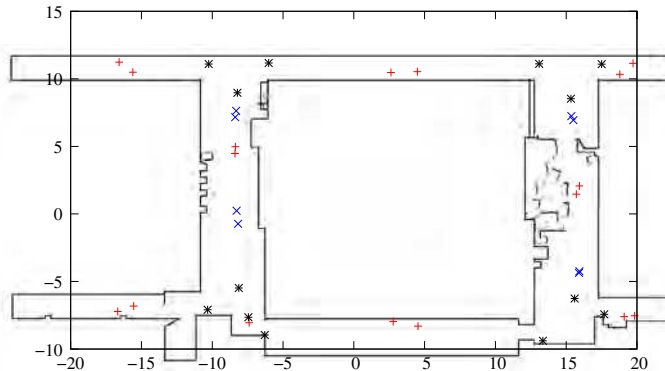


Fig. 6. Location distribution over the map. Corridors: +; Halls: x; Crossings: *

In spite of the degree of symmetry of the environment, the spatial configuration of the obtained locations does not show the same degree of symmetry. This is due to the fact that robot's and humans' perception differ from each other, and since the robot navigates according to a desired compass heading, depending on its orientation it makes the same physical place correspond to several nodes in the topological representation.

9. Experiments in the real robot/environment system

The simulation experiments showed that the proposed approach can solve the stated problem. To test the robustness of the approach experiments were extended to the real robot-environment system. The robot Tartalo—a *PeopleBot* robot from *MobileRobots* equipped with a Canon VCC5 monocular PTZ vision system, a Sidel LMS laser, a TCM2 compass and several sonars and bumpers—has been used for the empirical evaluation of the mapping system developed. But instead of relying on raw odometry information, two odometry correction methods were tested to smooth the positioning error:

- *Laser stabilised odometry* (by means of the LODO driver provided by *Player*). Laser data is used to correct the raw odometry estimate that once corrected exhibits a drift rate that is an order of magnitude less than the rate observed using pure odometry (Howard, 2005).
- Compass based odometry (CODO), where compass heading is used to correct raw odometric poses.

Experiments were performed in the third floor of the Faculty of Computer Sciences. On the left of Figure 7 shows the path completed by the robot (according to compass based odometry) and on the right shows the evolution of the number of nodes over time (s) for the different positioning methods. Clearly, the compass odometry obtained with the proposed approach offers the most precise position information.

On the other hand, Figure 8 shows the distribution of locations of the different nodes obtained from the run performed by the robot using CODO (Figure 7). As mentioned in the previous section, the difference in perception explains the fact that the number of nodes acquired by robot and humans differ from each other.

And, as expected, the number of nodes is higher when the mapping is performed by the robot because of its perception of the environment and its positioning error. However, although the number of junction nodes identified is higher in the real run, this is mainly due to the people and furniture the robot comes across, which produce nodes that lead to any number of alternative paths. However, after an exploration of about an hour and a half (more than 500 meters), the robot was able to close the loop and to recognise several times the final location as the starting one, thus confirming the suitability of the proposed approach.

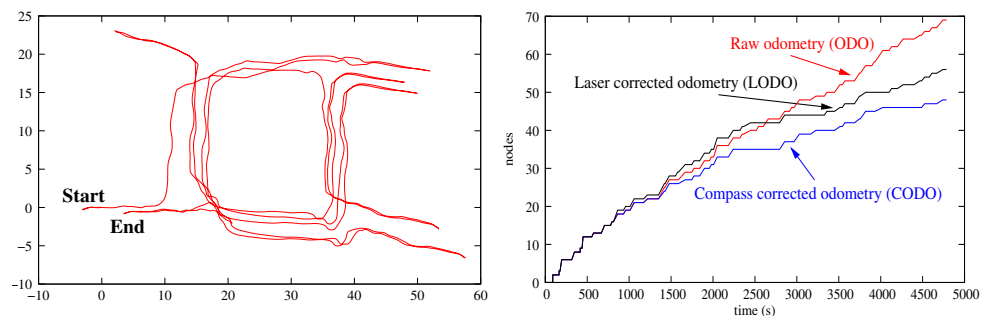


Fig. 7. Left: Robot's path (CODO). Right: Evolving number of nodes

As mentioned earlier, the experiments performed in simulation cannot be directly compared with the experiments with the real robot; the simulated sensor readings produce nodes with different characteristics specially when junction nodes are identified. Hence, the path produced by the exploration strategy in simulation differs from the path executed by the real robot. However, it is interesting to compare the evolution of the learning process using exact odometry with the evolving number of nodes when the odometry is corrected using the compass sensor. The map obtained simulating ideal odometry converged to 38 nodes and the map obtained by the robot after 4500 seconds contained 48 nodes (see Figure 9).

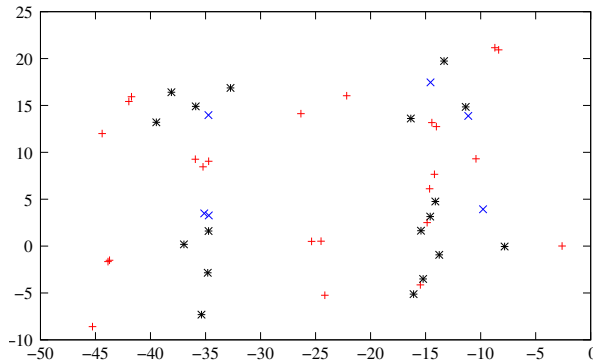


Fig. 8. Location distribution over the map. Corridors: +; Halls: x; Crossings: *

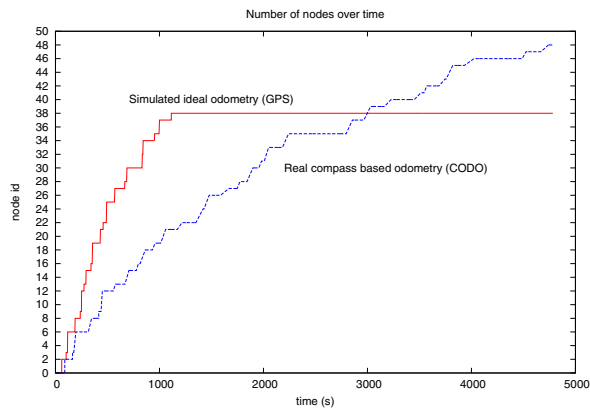


Fig. 9. Comparison with ideal odometry

10. INCA for localisation

During the previous experiments the learning process was not stopped once the loop was closed. This methodological criterion was chosen to assess the appropriateness of the approach, and as a result, there was a slow increase in the number of nodes over time mainly due to odometry error. However, in practical terms the map learning process can be stopped and then use the learnt map for localisation purposes.

The experiments described in this section were carried out to measure the usefulness of the acquired map for localisation. In this occasion, instead of a non-stop learning process, a criterion was set so that the generation of the map would stop once a certain number of nodes was included. Once the procedure reaches this value, no more nodes are allowed to be created and hence, classification rule 6 (Section 3.2) gives the closest node according to the available data. In this manner, after the map is completed the robot continues moving according to its exploration strategy while the mentioned rule gives its localisation. It is worth to mention that classification rule 6 is equivalent to the distance based classifier introduced in (Cuadras, 1992).

Experiments were conducted both in simulation and in the real robot/environment system.

10.1 Simulated experiments

Once more the *Stage* simulator was used to simulate the robot and its environment. The criterion to stop the learning process was established in 38 nodes, which was the number of nodes the map converged to when simulating the mapping process with ideal odometry. Two experiments were carried out in the simulator:

- Ideal odometry (GPS). Figure 10 shows the journey together with the spatial node configuration learnt whereas Figure 11 shows the results of the mapping and localisation process, and thus the identified set of nodes over time. The mapping process lasted about 1100 seconds and the fact that no error occurred during the localisation phase (seconds 1100-12000) confirmed that INCA is a valid approach also for localisation. Once the map has been generated, the trajectory of the robot is randomly decided at run-time. The resulting unpredictability means that instead of following a static route, the robot will randomly select the orientation at each junction. As a consequence, the robot does not produce repeatable sequences of nodes in the path, but the probability that it will revisit the whole set of nodes increases.
- Laser corrected odometry (LODO). A new experiment was conducted applying the default odometry error value defined in *Player/Stage* and applying the LODO driver to correct the odometry. Figure 12 shows the journey together with the spatial node configuration learnt whereas Figure 13 shows the results of the mapping and localisation process, and thus the identified set of nodes over time.

Table 2 shows the path patterns extracted from plot in Figure 13(a). Again, their associated node sequence, the time interval and the label used in the plot to represent each pattern is included.

The identified patterns concentrate in the first part of the plot (seconds 2000 to 12000). As times goes by the extracted paths are shorter due to localisation failures and the task becomes extremely difficult from second 12000 and there on. Although an odometry correction method is applied, the accumulating error severely affects the localisation of the robot. The type of error remaining after the LODO correction procedure produces a rotation on the robot's trajectory (see for instance Figure 12(a)) and thus, a misclassification of nodes with different orientations assigned. This effect was detected in the sequences labelled as *c0 – c1* in Table 2. Chain P17, B34, P38 should have been P31, B34, P38. Node P31 with assigned orientation SN was misclassified as node P17 with assigned orientation WE.

Note that both procedures produced the same configuration of nodes, 17 corridors, 13 crosses and 8 halls, although their positional information differed due to odometry values.

10.2 Experiments in the real robot/environment system

A second set of experiments were carried out with the real robot. This time the node threshold was established in 44 nodes.

Figure 14 shows the robot's path and the obtained node distribution using laser corrected odometry values (LODO).

Figure 15 shows the robot's path and the obtained node distribution using compass corrected odometry values (CODO). And Figure 16 shows the mapping process, and the localisation over time for both, LODO and CODO.

The results were disappointing but confirmed what the simulated experiments showed for the LODO case. Although the robot localises properly for about 2000 seconds, afterwards the

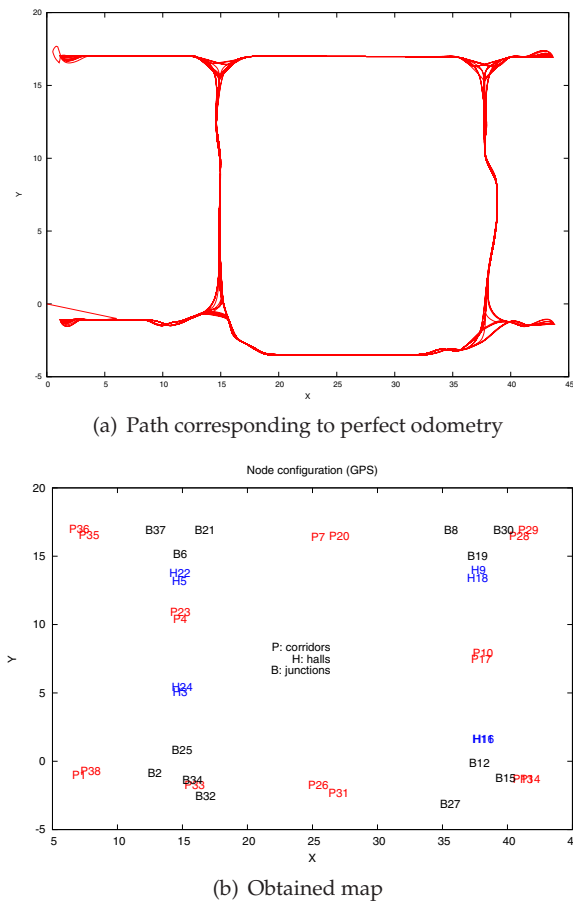


Fig. 10. *Stage* (GPS): robot's path and the obtained map

localisation starts to degrade. It is not possible to extract valid path patterns from the plots in Figure 16. Both the LODO and CODO methods are insufficient for long term localisation.

Looking at the robot's paths drawn in Figures 14(a) and 15(a), it can be stated that:

- When using the LODO correction method, the error accumulates more slowly but the error occurs in x , y and θ coordinates. According to LODO odometry, the path rotates over time. While the error is maintained within a certain range the rotation angle is small and the localisation process works correctly. Afterwards, and due to the high dependency the approach has in nominal orientations the system starts to fail and no correspondences are found.
- When using the CODO correction method, only x and y values are affected. θ value is obtained from an absolute reference value and hence, the error is not accumulative. This produces a diagonal shift on the drawn path over time. This shift led to the misclassification of the lower corridors as if they were the upper ones. Oddly, the upper corridors were always well identified.

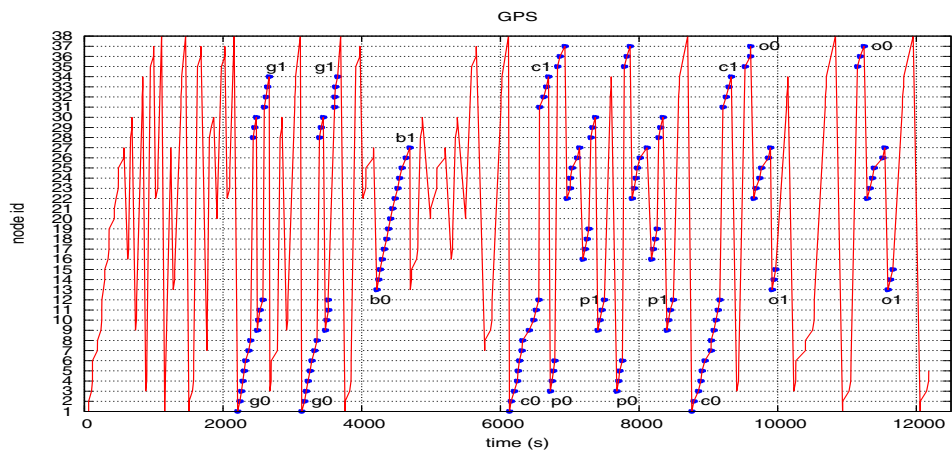


Fig. 11. Stage (GPS): node identification over time

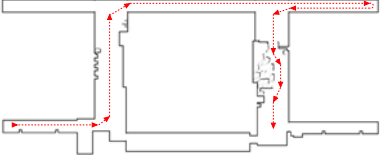
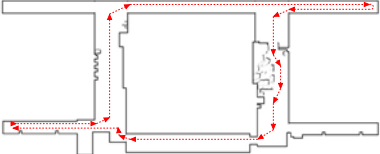

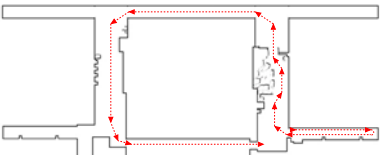
Robot's path	node sequence	time stamp	label
	P1, B2, H3, P4, H5, B6, P7, B8, P28, P29, B30, H9, P10, H11, B12	2128-2391, 2955-3227	g0-g1
	P1, B2, H3, P4, H5, B6, P7, B8, P28, P29, B30, H11, P10, H11, B12, P17*, B34, P38	5767-6217, 6301-6916, 6981-7606	c0-c1
	P35, P36, B37, H22, P23, H24, B25, P26, B27	3565-3781, 9130-9413, 11322-11627	m0-m1
	P13, P14, B15, H16, P17, H18, B19, P20, B21, H22, P23, H24, B25, P26, B27	3821-4169	b0-b1

Table 2. LODO: extracted path patterns (*: localisation error)

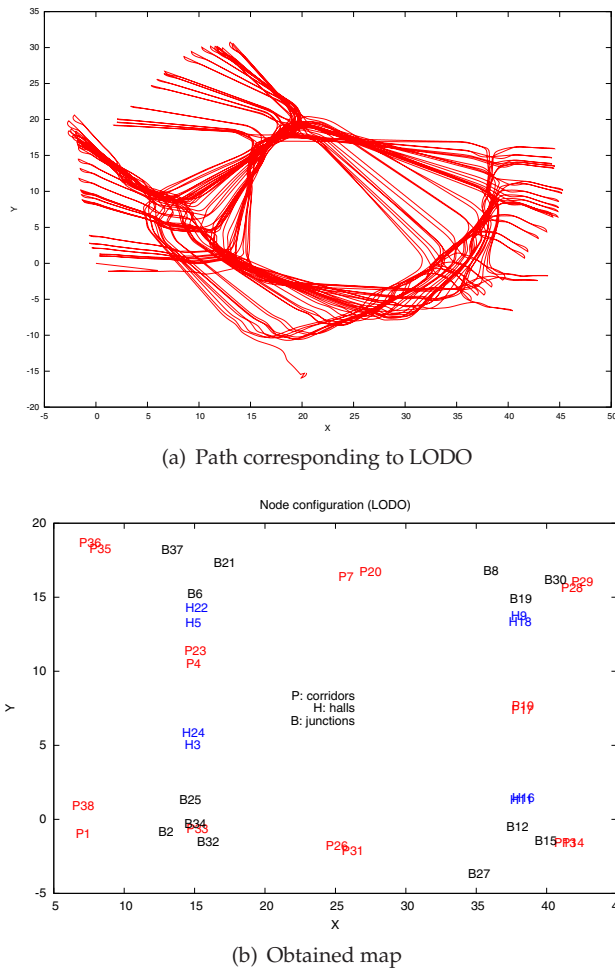


Fig. 12. *Stage* (LODO): robot's path and the obtained map

An intuitive way of coping with this problem is to modify the positional values of the nodes each time they are revisited. Instead of keeping the acquired node information unaltered, during the localisation phase the contents of the nodes can be updated when a positive match occurs.

A last experiment was performed with the robot using CODO to correct the odometry to measure the effect of updating the contents of the node. This choice was made because of the lack of accumulated error in orientation values. Figure 17 shows the acquired map after reaching the maximum number of nodes (established in 39 nodes). The different scales of these two maps reflect the magnitude of the accumulated error in the x and y coordinates over time. Figure 18 shows the evolution of the localisation system over time.

Table 3 shows the path patterns extracted from plot in Figure 18. Again, their associated node sequence, the time interval and the label used in the plot to represent each pattern is included.

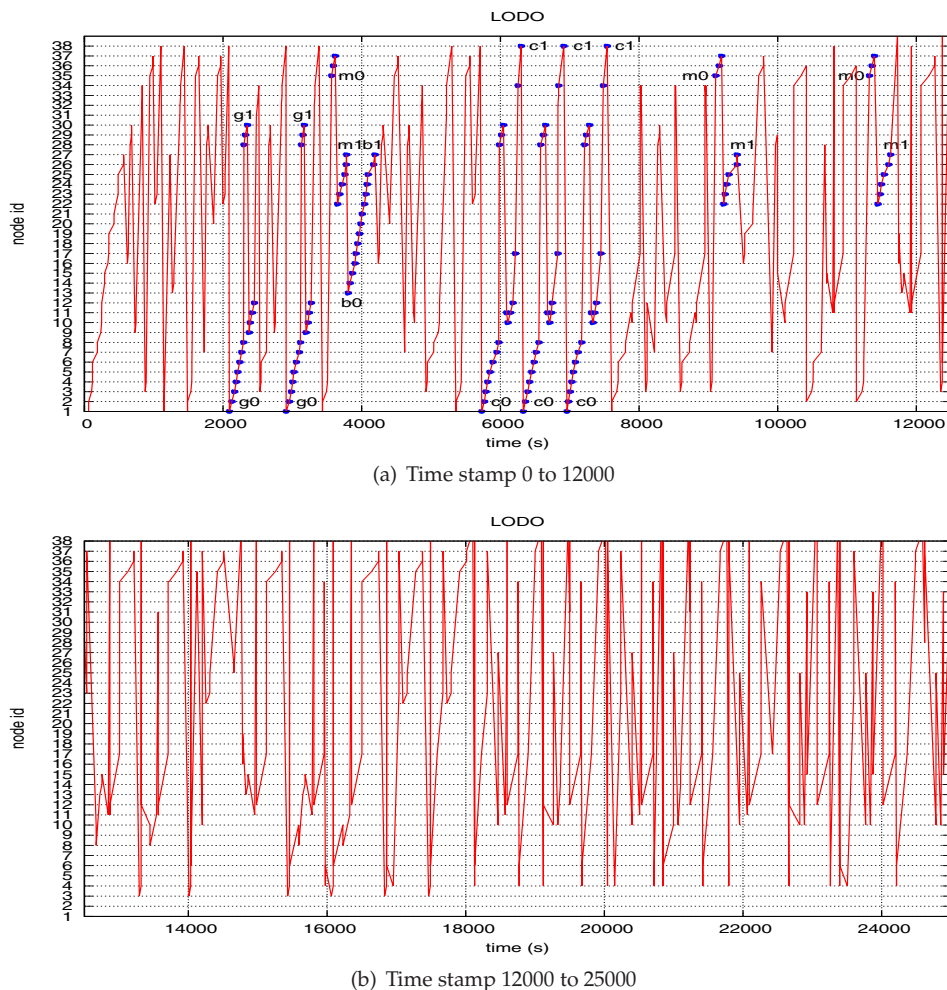


Fig. 13. *Stage (LODO): node identification over time*

The localisation process last until the robot run out of batteries and only one location was misclassified. As mentioned in Section 6, some parameters need to be adjusted for INCA to function properly. The value u_M deeply influences the acceptable deviations from nodes' (x, y) locations. A small u_M value produces failures on loop-closings because of the odometry error. On the contrary, setting u_M to a high value produces that close areas with the same signatures remain indistinguishable. This effect was detected once during the last localisation experiment carried out. Node H30 was wrongly identified as node H32 and thus, the sequence H32, P31, H32 of time stamp 1700 should have been H30, P31, H32. Notice that nodes H30 and H32 are separated by a short corridor labelled as P31. Summarising, upgrading node information made the developed system valuable for localisation in spite of odometry error.

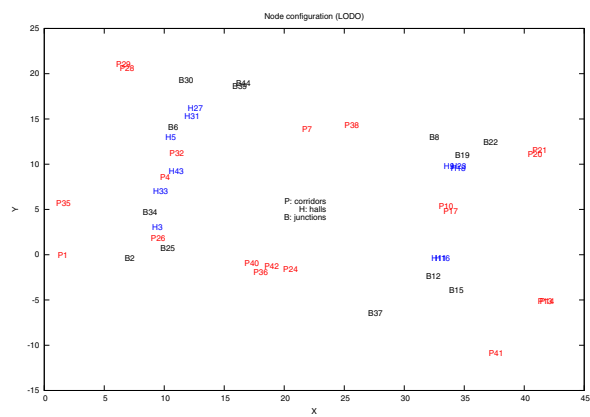
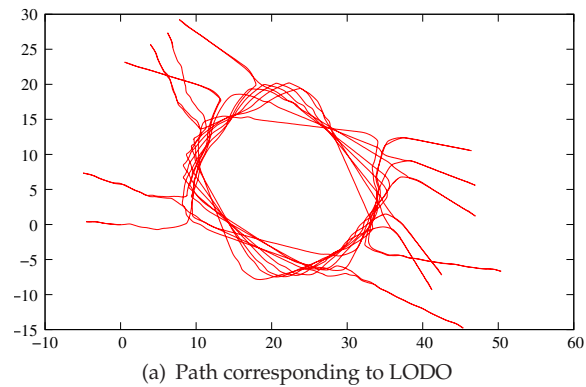
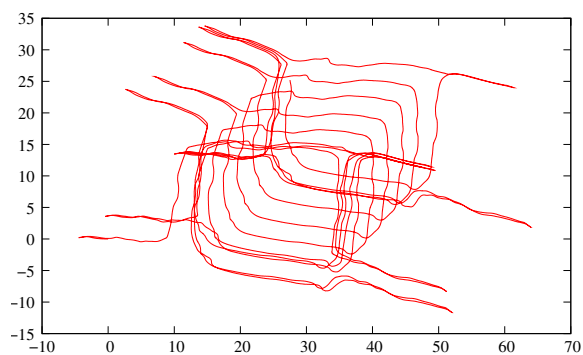
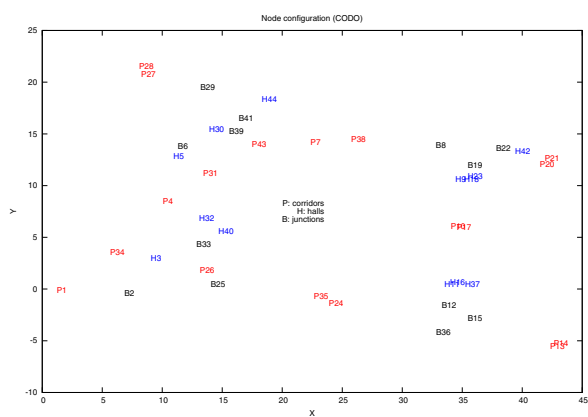


Fig. 14. Tartalo (LODO): robot's path and the obtained map

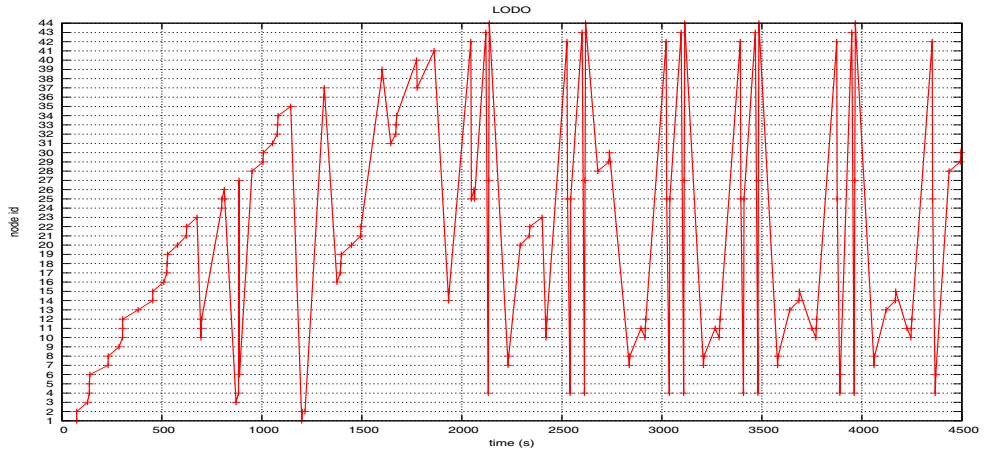


(a) Path corresponding to compass odometry

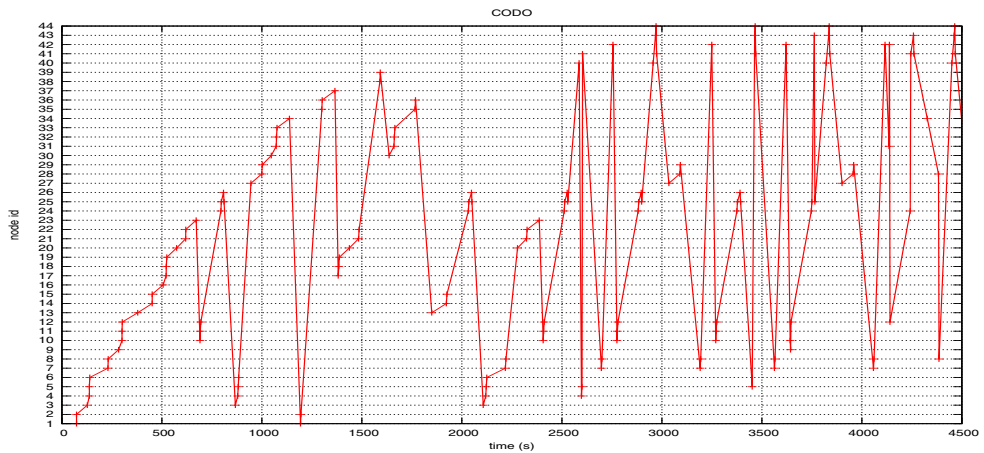


(b) Obtained map

Fig. 15. Tartalo (CODO): robot's path and the obtained map

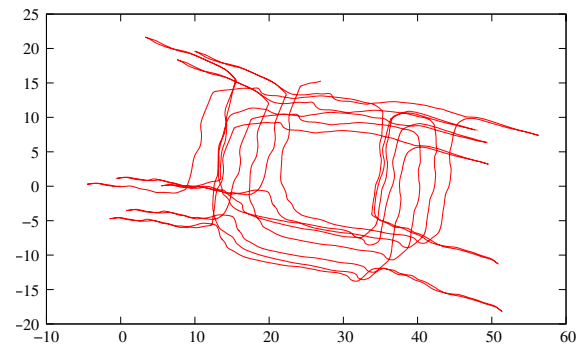


(a) Laser corrected odometry

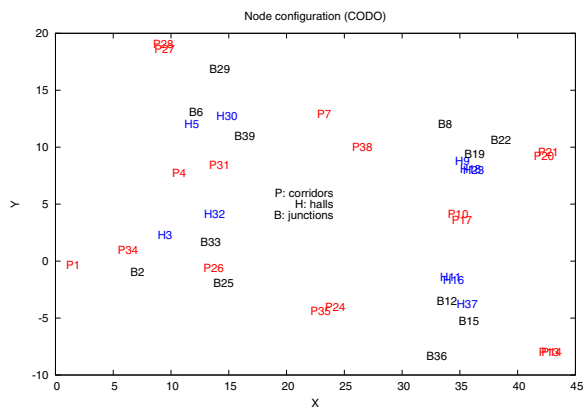


(b) Compass corrected odometry

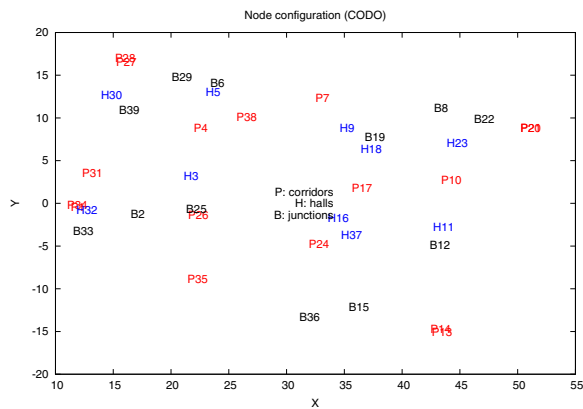
Fig. 16. Tartalo(LODO and CODO): node identification over time. Labels are set to show extracted path patterns



(a) Robot's path



(b) Original map



(c) Upgraded map

Fig. 17. Tartalo (CODO with adaptive node location): robot's path and the obtained map

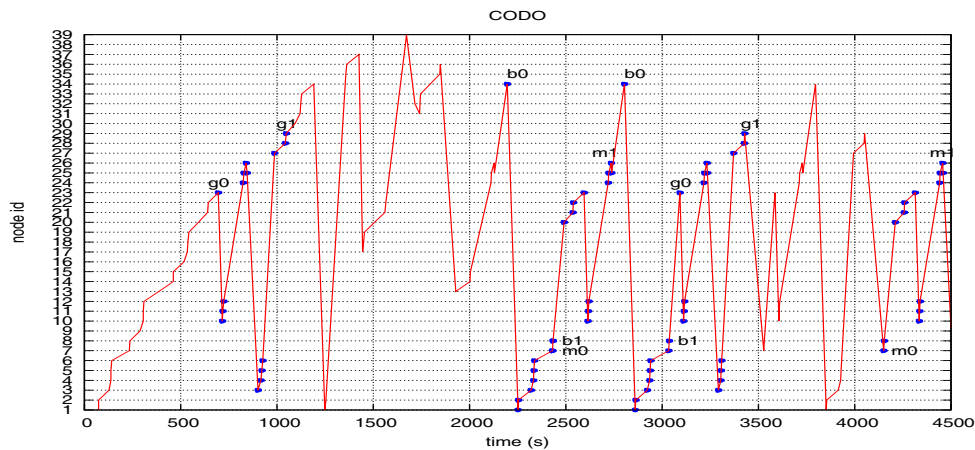


Fig. 18. Tartalo (CODO with adaptive node location): node identification over time

Robot's path	node sequence	time stamp	label
	H23, P10, H11, B12, P24, B25, P26, B25, H3, P4, H5, B6, P27, P28, B29	643-1048, 3092-3428	g0-g1
	P34, P1, B2, H3, P4, H5, B6, P7	2195-2430, 2804-3034	b0-b1
	P7, B8, P20, P21, B22, H23, P10, H11, B12, P24, B25, P26, B25	2430-2737, 4150-4459	m0-m1

Table 3. CODO: extracted path patterns

11. Conclusions

In this chapter a new approach for incremental topological map construction was presented. A statistical test called INCA was used to this end, combined with a data sampling approach which decided if a topological node found by the robot had already been visited by it. The method was integrated in a behaviour-based control architecture and tested also for localisation purposes.

To measure the adequateness of the approach the map acquisition was performed non-stop until the robot run out of batteries. Afterwards, the experiments were repeated but once the number of nodes in the map reached a given threshold, the learning step was finished and the acquired map was used for localisation purposes.

However, INCA also suffers from odometry error. Of the two error correction methods used in the present work, LODO and CODO, compass corrected odometry was better suited for the developed navigation approach. A last experiment was carried out using CODO and modifying the contents of the acquired nodes each time a location was revisited. The type of error remaining after CODO facilitated the upgrade of the nodes' locations and improved drastically the localisation process.

The experiments conducted confirmed INCA based mapping and localisation as a valid approach and that BB systems can be provided with automatic map acquisition mechanisms. To improve the efficiency of the automatic map acquisition system, when looking for correspondences the use of their associated probability value should be studied.

The criteria for stopping the learning process, i.e. the maximum number of nodes should be revised. Given that it is not possible to know a priori the number of nodes, the map should be closed when no more alternative ways remain unvisited in the junction nodes.

Some aspects of the implementation of INCA should be improved and more experiments should be conducted in a systematic manner in order to better identify the advantages and drawbacks of the test.

New local navigation strategies and several landmark identification modules need to be incorporated to increase the granularity of the environment in order to reach more interesting goals than halls and corridors, such as offices and laboratories. Adding more topological nodes would allow the generalisation of the experiments to different environments, and the comparison with other approaches. The first step should be to integrate the door identification and door crossing modules already developed, and to enrich the behaviour associated to several nodes with door crossing abilities, and a wall following behaviour. These two modules would help to cover the perimeter of small rooms and improve the exploration strategy.

Nothing has been said about planning. Up to now, the proposed modifications were tested using an exploration strategy. The overall map should be used for commanding the robot to fulfil a concrete goal and thus, to reach concrete locations.

12. References

- Arenas, C. & Cuadras, C. M. (2002). Some recent statistical methods based on distances, *Contributions to Science* 2: 183–191.
- Bar-Hen, A. (2001). Preliminary tests in linear discriminat analysis, *Statistica* 4: 585–593.
- Brooks, C. & K. Iagnemma, K. (2009). Visual detection of novel terrain via two-class classification, *Proceedings of the ACM symposium on Applied Computing (SAC)*, ACM, New York, NY, USA, pp. 1145–1150.

- Brooks, R. A. (1986). A robust layered control system for a mobile robot, *IEEE Journal of robotics and automation* RA-26: 14–23.
- Brooks, R. A. (1989). A robot that walks: emergent behaviors from a carefully evolved network, *Technical Report AI MEMO 1091*, MIT.
- Brooks, R. A. & Connell, J. H. (1986). Asynchronous distributed control system for a mobile robot, *Proceedings of the SPIE's Cambridge Symposium on Optical and Optoelectronic Engineering*, pp. 77–84.
- Chen, C. & Wang, H. (2006). Appearance-based topological bayesian inference for loop-closing detection in a cross-country environment, *International Journal of Robotic Research* 25(10): 953–983.
- Connell, J. H. (1990). *Minimalist mobile robotics. A colony-style architecture for an artificial creature*, Academic Press, inc.
- Cuadras, C. M. (1992). Some examples of distance based discrimination, *Biometrical Letters* pp. 3–20.
- Cuadras, C. M. & Fortiana, J. (1995). A continuous metric scaling solution for a random variable, *Journal of Multivariate Analysis* 32: 1–14.
- Cuadras, C. M. & Fortiana, J. (2000). The importance of geometry in multivariate analysis and some applications, *Statistics for the 21st Century*, Marcel Dekker, New York, pp. 93–108.
- Cuadras, C. M., Fortiana, J. & Oliva, F. (1997). The proximity of an individual to a population with applications in discriminant analysis, *Journal of Classification* 14: 117–136.
- Fraundorfer, F., Engels, C. & Nistér, D. (2007). Topological mapping, localization and navigation using image collections, *Intelligent Robots and Systems (IROS)*, pp. 3872–3877.
- Gower, J. C. (1985). *Encyclopedia of Statistical Sciences*, Vol. 5, John Wiley & Sons, New York, chapter Measures of similarity, dissimilarity and distance, pp. 397–405.
- Hempstalk, K., Frank, E. & Witten, I. H. (2008). One-class classification by combining density and class probability estimation, *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, Springer-Verlag, Berlin, Heidelberg, pp. 505–519.
- Howard, A. (2005). Multi-robot simultaneous localization and mapping using particle filters, *In Robotics: Science and Systems*, pp. 201–208.
- Irigoin, I. & Arenas, C. (2008). INCA: New statistic for estimating the number of clusters and identifying atypical units, *Statistics in Medicine* 27(15): 2948–2973.
- Jauregi, E., Irigoin, I., Sierra, B., Lazkano, E. & Arenas, C. (2011). Loop-closing: a typicality approach, *Robotics and Autonomous Systems* 59: 218–227.
- Mallot, H. A. & Franz, M. A. (2000). Biomimetic robot navigation, *Robotics and Autonomous System* 30: 133–153.
- Manevitz, L. M. & Yousef, M. (2002). One-class SVMs for document classification, *J. Mach. Learn. Res.* 2: 139–154.
- Manevitz, L. & Yousef, M. (2007). One-class document classification via neural networks, *Neurocomput.* 70(7-9): 1466–1481.
- Matarić, M. (1990). *A distributed model for mobile robot environment-learning and navigation*, Master's thesis, MIT Artificial Intelligence Laboratory.
- McDonald, L. L., Lowe, V. W., Smidt, R. K. & Meister, K. A. (1976). A preliminary test for discriminant analysis based on small samples, *Biometrics* 32: 417–422.

- Olson, E. (2009). Recognizing places using spectrally clustered local matches, *Robotics and Autonomous System* 57: 1157–1172.
- Rao, C. R. (1962). Use of discriminant and allied functions in multivariate analysis, *Sankhya-Serie A* 24: 149–154.
- Rao, C. R. (1982). Diversity: its measurement, decomposition, apportionment and analysis, *Sankhyā. The Indian Journal of Statistics, Series A* 44: 1–22.
- Sánchez-Yáñez, R. E., Kurmyshev, E. V. & Fernández, A. (2003). One-class texture classifier in the CCR feature space, *Pattern Recognition Letters* 24(9-10): 1503–1511.
- Se, S., Lowe, D. G. & Little, J. J. (2005). Vision-based global localization and mapping for mobile robots, *IEEE Transactions on Robotics* 21: 364–375.
- Tardós, J. D., Neira, J., Newman, P. M. & Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data, *The International Journal of Robotics Research* 21(4): 311–330.
- Tax, D. (2001). *One-class classification; Concept-learning in the absence of counter-examples*, PhD thesis, Delft University of Technology.
- Thrun, S. (1999). Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intelligence* pp. 21–71.
- Thrun, S. & Bücken, A. (1996a). Integrating grid-based and topological maps for mobile robot navigation, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 944–950.
- Thrun, S. & Bücken, A. (1996b). Learning maps for indoor mobile robot navigation, *Technical report*, Carnegie Mellon University.
- Thrun, S., Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, MIT Press.
- Wang, Q. & Lopes, L. S. (2005). *Emerging Solutions for Future Manufacturing Systems*, Vol. 159, Springer Boston, chapter One-Class Learning for Human-Robot Interaction, pp. 489–498.
- Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I. & Tardós, J. (2009). A comparison of loop closing techniques in monocular SLAM, *Robotics and Autonomous System* 57: 1157–1172.



Mobile Robots - Current Trends

Edited by Dr. Zoran Gacovski

ISBN 978-953-307-716-1

Hard cover, 402 pages

Publisher InTech

Published online 26, October, 2011

Published in print edition October, 2011

This book consists of 18 chapters divided in four sections: Robots for Educational Purposes, Health-Care and Medical Robots, Hardware - State of the Art, and Localization and Navigation. In the first section, there are four chapters covering autonomous mobile robot Emmy III, KCLBOT - mobile nonholonomic robot, and general overview of educational mobile robots. In the second section, the following themes are covered: walking support robots, control system for wheelchairs, leg-wheel mechanism as a mobile platform, micro mobile robot for abdominal use, and the influence of the robot size in the psychological treatment. In the third section, there are chapters about I2C bus system, vertical displacement service robots, quadruped robots - kinematics and dynamics model and Epi.q (hybrid) robots. Finally, in the last section, the following topics are covered: skid-steered vehicles, robotic exploration (new place recognition), omnidirectional mobile robots, ball-wheel mobile robots, and planetary wheeled mobile robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

E. Jauregi, I. Irigoien, E. Lazkano, B. Sierra and C. Arenas (2011). Robotic Exploration: Place Recognition as a Typicality Problem, *Mobile Robots - Current Trends*, Dr. Zoran Gacovski (Ed.), ISBN: 978-953-307-716-1, InTech, Available from: <http://www.intechopen.com/books/mobile-robots-current-trends/robotic-exploration-place-recognition-as-a-typicality-problem>

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.